



# UNITED STATES PATENT AND TRADEMARK OFFICE

80  
UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/845,750	04/30/2001	Walter Dixon III	345708004US	3459
28062	7590	08/09/2005	EXAMINER	
BUCKLEY, MASCHOFF, TALWALKAR LLC				ZHEN, LI B
5 ELM STREET				
NEW CANAAN, CT 06840				
ART UNIT		PAPER NUMBER		
		2194		

DATE MAILED: 08/09/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	09/845,750	DIXON ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	Li B. Zhen	2194	

**— The MAILING DATE of this communication appears on the cover sheet with the correspondence address —**

**Period for Reply**

**A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.**

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1) Responsive to communication(s) filed on 20 May 2005.

2a) This action is **FINAL**.                                    2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4) Claim(s) 3-8,10-12,14-21,23-32,36-40,44 and 46-67 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 3-8,10-12,14-21,23-32,36-40,44 and 46-67 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All    b) Some \* c) None of:

- Certified copies of the priority documents have been received.
- Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
- Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.

4) Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.

5) Notice of Informal Patent Application (PTO-152)

6) Other: \_\_\_\_\_.

## **DETAILED ACTION**

1. Claims 3 – 8, 10 – 12 and 14 – 21, 23 – 32, 36 – 40, 44 and 46 – 67 are pending in the current application.

### ***Claim Objections***

2. Claim 59 is objected to because of the following informalities: “sub-applications if performed” [emphasis added, line 2]. Appropriate correction is required.

### ***Claim Rejections - 35 USC § 102***

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. **Claims 3 – 8, 10 – 12, 14 – 16 and 20, 21, 23 – 32, 36 – 40, 44 and 46 - 67 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,813,769 to Limprecht et al. [hereinafter Limprecht].**

5. As to claim 3, Limprecht teaches a method in a computer system for dispatching requests to perform services to sub-applications [server application components 86 that are hosted in the execution environment 80 of the ASP 90 implement the business logic

of a server application; col. 7, line 65 – col. 8, line 10] that use different logic models [e.g. COM or CORBA; col. 8, lines 11 – 23] the method comprising:

providing a context for the sub-applications [maintains a component context object 138 associated with the server application component; col. 10, lines 5 – 37]  
receiving a request from a client computer to perform a service [client program 134 calls a member function of the server application component 86; col. 19, lines 63 – 67]; and

for a plurality of sub-applications [component pool 180, Fig. 6; col. 21, lines 1 – 15],

determining whether the received request should be dispatched to the sub-application [at step 220, the transaction server executive 80 first checks whether there is an instance of the server application component in the component pool 180; col. 21, lines 1 – 15]; and

when it is determined that the request should be dispatched to the sub-application [At step 226, the transaction server executive 80 then passes the client program's call; col. 21, lines 38 – 43], invoking a service routine of the sub-application passing the request [initiates processing the call by the appropriate function of the server application component; col. 21, lines 38 – 43]

whereby the sub-applications share the provided context [inherit context from the component context object (i.e., the component context objects created for the new components have the same context properties, including client id, activity id and transaction, as the original component context object); col. 14, lines 3 – 25];

wherein the determining includes determining whether a match criteria [step 223, issuing a QueryInterface to check whether the sever application component supports IObjectControl interface, Fig. 7B; col. 21, lines 25 – 38] for the sub-application matches the received request [calls QueryInterface to determine that the server application component provides the IObjectControl interface; col. 19, lines 1 – 17].

6. As to claim 4, Limprecht teaches the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL [col. 6, lines 38 – 65].
7. As to claim 5, Limprecht teaches suppressing the invoking of additional service routines when an invoked service routine returns an indication to suppress the invoking of additional service routines [col. 4, lines 19 – 34].
8. As to claim 6, Limprecht teaches suppressing the invoking of additional service routines when an invoked service routine responds to the received request [col. 17, line 62 – col. 18, line 10].
9. As to claim 7, Limprecht does not teach an invoked service routine performs user authentication and indicates to suppress invoking of additional service routines when a user cannot be authenticated. However, this is well known in the art (such as the authentication module in a web server that must authenticate a user before allowing the server to return a page).

10. It would have been obvious to one of ordinary skill in the art at the time of the invention to use such an authentication scheme with the system of Limprecht in order to provide security for the accessed services.

11. As to claim 8, Limprecht does not teach an invoked service routine logs the received request. However, the logging of received requests is well known in the art (access logs for a web server).

12. It would have been obvious to one of ordinary skill in the art at the time of the invention to invoke service routine log received requests in the system of Limprecht in order to decrease the costs associated with debugging the system.

13. As to claim 10, Limprecht does not disclose transforming the received request from one protocol to another. However, such transformations are well known in the art (WAP gateways that transform between HTML and WMQ).

14. It would have been obvious to one of ordinary skill in the art at the time of the invention to use such a transforming between protocols in the system of Limprecht in order to allow access to different clients.

15. As to claim 11, Limprecht teaches for each of a plurality of sub-applications [component pool 180, Fig. 6; col. 21, lines 1 – 15], retrieving initialization parameters for the sub-application [server application component passes the parameters; col. 23, lines 18 – 36];

retrieving an indication of a class for the sub-application ["rclsid" is a class identifier that specifies the new server application component to be created; col. 23, lines 18 – 36]; and

instantiating an instance of the class with the retrieved initialization parameters [create a new server application component; col. 23, lines 18 – 36].

16. As to claim 12, Limprecht does not teach that the match criteria is in a configuration file for the sub-application. However, the use of configuration files is well known in the art.

17. It would have been obvious to one of ordinary skill in the art at the time of the invention to store the match criteria for the sub-applications in configuration files in the system of Limprecht to prevent having to re-enter this information whenever the system is restarted.

18. As to claims 14, 15 and 16, Limprecht does not teach an interaction-based model, an action-view model or a workflow-based model. However, these logic models are all well known in the art.

19. It would have been obvious at the time of the invention to use these logic models in the sub-applications of Limprecht in order to use the architecture that is most appropriate for handling different requests.

20. As to claim 20, Limprecht teaches a computer system for dispatching HTTP [col. 6, lines 38 – 65] requests to sub-applications [col. 7, line 65 – col. 8, line 10], comprising:

a configuration file having a class, initialization parameters [server application component passes the parameters; col. 23, lines 18 – 36], and a match criteria associated with the sub-applications [col. 19, lines 1 – 17];

an initialization component that instantiates an object of the class for each sub-application in the configuration file [create a new server application component; col. 23, lines 18 – 36], the instantiated object being initialized with the initialization parameters for the sub-application [col. 23, lines 18 – 36] and being provided with a context object [col. 14, lines 3 – 25], the context object being shared by the instantiated objects so that the sub-applications share a common context [inherit context from the component context object (i.e., the component context objects created for the new components have the same context properties, including client id, activity id and transaction, as the original component context object); col. 14, lines 3 – 25]; and

a dispatcher that receives HTTP requests from client computers [client program 134 calls a member function of the server application component 86; col. 19, lines 63 – 67] and, when the received HTTP request matches a match criteria [step 223, issuing a QueryInterface to check whether the sever application component supports IObjectControl interface, Fig. 7B; col. 21, lines 25 – 38] of a sub-application [calls QueryInterface to determine that the server application component provides the IObjectControl interface; col. 19, lines 1 – 17], invokes a service routine of the

instantiated object of the class associated with the sub-application [col. 21, lines 38 – 43].

21. As to claim 21, this is rejected for the same reasons as claims 4 and 11 above.
22. As to claim 23, this is rejected for the same reasons as claims 5 and 11 above.
23. As to claim 24, this is rejected for the same reasons as claims 6 and 11 above.
24. As to claim 25, this is rejected for the same reasons as claims 11 and 14 above.
25. As to claim 26, this is rejected for the same reasons as claim 11 and 15 above.
26. As to claim 27, Limprecht teaches each of the sub-applications implement the same interface [col. 8, lines 11 – 23].
27. As to claim 28, Limprecht teaches a computer system for processing request messages, comprising:  
a plurality of sub-applications [maintains a component context object 138 associated with the server application component; col. 10, lines 5 – 37] forming an application, a sub-application having a match criteria indicating when the sub-application should process a request [col. 19, lines 1 – 17 and col. 21, lines 1 – 15] and

having a service routine to invoke when the match criteria [step 223, issuing a QueryInterface to check whether the sever application component supports IObjectControl interface, Fig. 7B; col. 21, lines 25 – 38] indicates that the sub-application should process the request [initiates processing the call by the appropriate function of the server application component; col. 21, lines 38 – 43], the sub-applications using disparate logic models [e.g. COM or CORBA; col. 8, lines 11 – 23]; a context for the application that is shared by the sub-applications [col. 10, lines 5 – 37]; and

a dispatcher that receives requests from client computers [col. 19, lines 63 – 67], evaluates the match criteria to identify which sub-applications have match criteria that match the requests [col. 19, lines 1 – 17], and invokes the service routines of the identified sub-applications wherein invoked sub-applications use the context [initiates processing the call by the appropriate function of the server application component; col. 21, lines 38 – 43].

28. As to claim 29, Limprecht teaches including an initialization component that instantiates an object of a specified class for each sub-application [create a new server application component; col. 23, lines 18 – 36].

29. As to claim 30, Limprecht teaches the initialization component accesses configuration information ["rclsid" is a class identifier that specifies the new server application component to be created; col. 23, lines 18 – 36] that specifies the class of

each sub-application and any initialization parameters for the sub-applications [server application component passes the parameters; col. 23, lines 18 – 36].

30. As to claim 31, Limprecht teaches a context object representing the context and wherein the initialization component provides the context object to each sub-application [inherit context from the component context object (i.e., the component context objects created for the new components have the same context properties, including client id, activity id and transaction, as the original component context object); col. 14, lines 3 – 25].

31. As to claim 32, Limprecht teaches each service routine is passed a request parameter [col. 23, lines 1 – 16] and returns a response parameter [col. 23, lines 40 – 57].

32. As to claim 36, this is rejected for the same reasons as claim 6, see the rejection to claim 6 above.

33. As to claim 37, Limprecht teaches a computer system for processing request messages, comprising:

a plurality of service means for servicing requests, the service means forming an application [maintains a component context object 138 associated with the server application component; col. 10, lines 5 – 37], each service means having a match

criteria indicating when the service means should be invoked [col. 19, lines 1 – 17 and col. 21, lines 1 – 15], the service means implementing different logic models [col. 8, lines 11 – 23]; and

dispatch means for receiving requests from client computers [col. 19, lines 63 – 67] evaluating match criteria to identify which service means have match criteria [step 223, issuing a QueryInterface to check whether the sever application component supports IObjectControl interface, Fig. 7B; col. 21, lines 25 – 38] that match the requests [col. 19, lines 1 – 17], and invoking the identified service means whereby the service means share a context that is common to the service means of the application [col. 21, lines 38 – 43].

34. As to claims 38 – 40, these are rejected for the same reasons as claims 29, 30 and 32 respectively, see the rejections to claims 29, 30 and 32 above.

35. As to claim 44, Limprecht teaches a computer-readable medium for controlling a computer system to dispatch requests to perform services to service routines, by a method comprising:

receiving a request from a client computer to perform a service [col. 19, lines 63 – 67]; and

for a plurality of service routines [component pool 180, Fig. 6; col. 21, lines 1 – 15], retrieving a match criteria for the service routine [col. 19, lines 1 – 17]; determining whether the received request matches the retrieved match

criteria [step 223, issuing a QueryInterface to check whether the sever application component supports IObjectControl interface, Fig. 7B; col. 21, lines 25 – 38]; when it is determined that the request matches the retrieved match criteria [col. 21, lines 38 – 43], invoking the service routine for processing of the received request whereby the service routines form an application and share a common context [initiates processing the call by the appropriate function of the server application component; col. 21, lines 38 – 43].

36. As to claims 46 – 48, these are product claims that correspond to method claims 4 – 6; note the rejections to claims 4 – 6 above, which also meet these product claims.

37. As to claims 49 – 53, Limprecht teaches all of the sub-applications or service means or service routines execute on the same server computer [component pool 180 is part of execution environment 82, which is on the server computer 84; see Figs. 2 and 6].

38. As to claims 54 – 58, Limprecht teaches a respective service routine is invoked for the request with respect to each of at least two of the sub-applications [IObjectContext interface 139 is used by the server application component 86 to create additional server application components, and to participate in the determination of transaction outcomes; col. 22, lines 60 – 67].

39. As to claim 59, Limprecht teaches the sub-applications are ordered [a list of instances held in the pool; col. 18, lines 25 – 48] and the invoking of the service routines of the at least two sub-applications is performed in the order of the sub-applications [col. 18, lines 25 – 48].

40. As to claim 60, Limprecht teaches the configuration file specifies an ordering of the sub-applications [col. 18, lines 25 – 48] and the dispatcher invokes the service routines of the instantiated objects of the classes [col. 23, lines 18 – 36] associated with the at least two sub-applications in the specified order [col. 18, lines 25 – 48].

41. As to claims 61 – 63, these are rejected for the same reasons as claims 59, 5 and 6 respectively, see the rejections to claims 59, 5 and 6 above.

42. As to claims 64 – 66, these are rejected for the same reasons as claims 59, 5 and 6 above.

43. As to claim 67, this is a product claim that corresponds to method claim 59; note the rejection to claim 59 above, which also meets this product claim.

***Claim Rejections - 35 USC § 103***

44. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**45. Claims 17 – 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Limprecht in view of U.S. Patent No. 6,718,535 to Underwood [cited in the previous office action].**

46. As to claim 17, Limprecht teaches the sub-applications form an overall application [server application components 86 that are hosted in the execution environment 80 of the ASP 90 implement the business logic of a server application; col. 7, line 65 – col. 8, line 10] but does not teach the provided context is an application-level context.

However, Underwood teaches the provided context is an application-level context [col. 324, lines 57 – 62].

47. It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of providing an application-level context as taught by Underwood to the invention of Limprecht because the context is used to control the scope of transactions performed by the sub-applications [col. 22, lines 46 – 56 of Underwood].

48. As to claim 18, Limprecht as modified teaches the sub-applications form an overall application that is web-based [col. 17, lines 60 – 67 of Underwood].

49. As to claim 19, Limprecht as modified teaches the request is received from a web-server environment [col. 17, lines 60 – 67 of Underwood].

***Response to Arguments***

50. Applicant's arguments filed May 20, 2005 have been fully considered but they are not persuasive. In response to the Non-Final office action dated February 23, 2005, applicant argues:

- (1) Limprecht does not teach receiving a request from a client computer to perform a service [p. 12, lines 13 – 16];
- (2) The call to QueryInterface is not a request from a client computer [p. 13, lines 8 – 9]; and
- (3) the IObejctControl interface is merely used to later deactivate the sever application component and has nothing to do with a match criteria for a sub-application or determining that the match criteria of the sub-application matches the request received from a client computer [p. 13, lines 9 – 12].

As to argument (1), examiner respectfully disagrees and notes that Limprecht teaches receiving a request from a client [col. 19, lines 63 – 67, see rejection to claim 3 above].

In response to argument (2), examiner agrees that QueryInterface is not a request from a client computer. However, examiner respectfully notes that the QueryInterface is called in response to a client request [step 207, Fig. 7A; col. 19, lines 60 – 67]. The QueryInterface call is used to determine whether a match criteria

[whether the sever application component supports IObjectControl interface] for the sub-application matches the received request [step 223, issuing a QueryInterface to check whether the sever application component supports IObjectControl interface, Fig. 7B; col. 21, lines 25 – 38].

As to argument (3), examiner respectfully disagrees and submits that QueryInterface is a call used to determine whether a match criteria [whether the sever application component supports IObjectControl interface] for the sub-application matches the received request [step 223, issuing a QueryInterface to check whether the sever application component supports IObjectControl interface, Fig. 7B; col. 21, lines 25 – 38]. In addition, IObjectControl interface is used to activate an instance of the server application component [step 224, Fig. 7B; col. 21, lines 30 – 38] and the client's request [program call] is processed [step 226, Fig. 7B; col. 21, lines 38 – 45].

### ***Conclusion***

51. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

52. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen  
Examiner  
Art Unit 2194

Ibz

  
MENG-AI T. AN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100